

WHAT IS CLAIMED IS:

- 1 1. A method for load balancing code execution, said
2 method comprising:
3 retrieving a code segment from a plurality of code
4 segments;
5 identifying a processor type from a plurality of
6 processor types to execute the code segment; and
7 loading the code segment into a processor that
8 corresponds to the identified processor type.
- 1 2. The method as described in claim 1 wherein the code
2 segment is byte code.
- 1 3. The method as described in claim 2 wherein the byte
2 code includes a byte code type, the byte code type
3 selected from the group consisting of Java, XML, HTML,
4 Shader, and Script.
- 1 4. The method as described in claim 2 further comprising:
2 compiling source code, the compiling resulting in the
3 byte code;
4 determining whether to store a pointer in a byte code
5 file, the pointer including a stored location that
6 corresponds to the byte code;
7 storing the pointer in the byte code file in response
8 to the determination;
9 storing the byte code at the stored location in
10 response to the determination; and
11 performing the retrieving using the pointer, wherein
12 the retrieving includes analyzing the stored location

13 and retrieving the byte code in response to the
14 analyzing.

1 5. The method as described in claim 1 wherein the
2 identifying includes analyzing the availability of
3 each of the plurality of processor types, and wherein
4 the analyzing includes retrieving a loading factor for
5 each of the plurality of processor types.

1 6. The method as described in claim 1 wherein the
2 identifying further comprises:
3 detecting one or more operations included in the code
4 segment; and
5 matching one or more of the operations with one of the
6 processor types from the plurality of processor types.

1 7. The method as described in claim 1 wherein the
2 identifying further comprises:
3 determining whether the code segment includes a
4 program directive corresponding to one of the
5 plurality of processors; and
6 matching one or more of the operations with one of the
7 processor types from the plurality of processor types
8 in response to the determination.

1 8. An information handling system comprising:
2 a plurality of processors;
3 a memory accessible by the processors;
4 one or more nonvolatile storage devices accessible by
5 the processors; and

6 a code execution load balancing tool for load
7 balancing code execution, the code execution load
8 balancing tool comprising software code effective to:
9 retrieve a code segment from a plurality of
10 code segments located in the memory;
11 identify a processor type from a plurality
12 of processor types to execute the code
13 segment; and
14 load the code segment into one of the
15 processors from the plurality of processors
16 that corresponds to the identified processor
17 type.

1 9. The information handling system as described in claim
2 8 wherein the code segment is byte code.

1 10. The information handling system as described in claim
2 9 wherein the byte code includes a byte code type, the
3 byte code type selected from the group consisting of
4 Java, XML, HTML, Shader, and Script.

1 11. The information handling system as described in claim
2 9 wherein the software code is further effective to:
3 compile source code, the compiling resulting in the
4 byte code;

5 determine whether to store a pointer in a byte code
6 file, the pointer including a stored location that
7 corresponds to the byte code;

8 store the pointer in the byte code file in the memory
9 in response to the determination;

10 store the byte code at the stored location in the
11 memory in response to the determination; and
12 perform the retrieving using the pointer, wherein the
13 retrieving includes analyzing the stored location and
14 retrieving the byte code from the memory in response
15 to the analyzing.

1 12. The information handling system as described in claim
2 8 wherein the identifying includes analyzing the
3 availability of each of the plurality of processor
4 types, and wherein the analyzing includes retrieving a
5 loading factor for each of the plurality of processor
6 types.

1 13. The information handling system as described in claim
2 8 wherein the software code is further effective to:
3 detect one or more operations included in the code
4 segment; and
5 match one or more of the operations with one of the
6 processor types from the plurality of processor types.

1 14. A computer program product stored on a computer
2 operable media for load balancing code execution, said
3 computer program product comprising:
4 means for retrieving a code segment from a plurality
5 of code segments;
6 means for identifying a processor type from a
7 plurality of processor types to execute the code
8 segment; and

9 means for loading the code segment into a processor
10 that corresponds to the identified processor type.

1 15. The computer program product as described in claim 14
2 wherein the code segment is byte code.

1 16. The computer program product as described in claim 15
2 wherein the byte code includes a byte code type, the
3 byte code type selected from the group consisting of
4 Java, XML, HTML, Shader, and Script.

1 17. The computer program product as described in claim 15
2 further comprising:

3 means for compiling source code, the compiling
4 resulting in the byte code;

5 means for determining whether to store a pointer in a
6 byte code file, the pointer including a stored
7 location that corresponds to the byte code;

8 means for storing the pointer in the byte code file in
9 response to the determination;

10 means for storing the byte code at the stored location
11 in response to the determination; and

12 means for performing the retrieving using the pointer,
13 wherein the retrieving includes analyzing the stored
14 location and retrieving the byte code in response to
15 the analyzing.

1 18. The computer program product as described in claim 14
2 wherein the identifying includes analyzing the
3 availability of each of the plurality of processor
4 types, and wherein the analyzing includes retrieving a

5 loading factor for each of the plurality of processor
6 types.

1 19. The computer program product as described in claim 14
2 wherein the identifying further comprises:
3 means for detecting one or more operations included in
4 the code segment; and
5 means for matching one or more of the operations with
6 one of the processor types from the plurality of
7 processor types.

1 20. The computer program product as described in claim 14
2 wherein the identifying further comprises:
3 means for determining whether the code segment
4 includes a program directive corresponding to one of
5 the plurality of processors; and
6 means for matching one or more of the operations with
7 one of the processor types from the plurality of
8 processor types in response to the determination.